art - Bug #2233

Support # 7559 (Closed): Memory check issues for Sirius A

SimpleMemoryChecker needs to be able to cope with different procfs formats on different SLF versions.

11/29/2011 04:11 PM - Christopher Green

Status: Start date: Closed 11/29/2011 Due date: **Priority:** Normal 09/30/2013 % Done: Assignee: Kyle Knoepfel 100% **Estimated time:** Category: Infrastructure 20.00 hours Target version: 1.13.00 Spent time: 20.00 hours Occurs In: **Experiment:** Scope: SSI Package: Internal art

Description

Chris Jones reported thusly:

Hi guys,

Are you providing the SimpleMemoryCheck service to your clients? If so, check the definition of linux_proc you are using. I just discovered that the types and even some of the fields do not match the present description of the linux proc file system. I'm guessing there was a change from SL4 to SL5 and we never caught the difference since VSize and RSS remained at their same location in the format.

Thought you should know,

Chris

We need to find out what changed between SLF4 and SLF5 (and potentially between SLF5 and SLF6), how to detect the changes and then act accordingly.

04/04/2014

Related issues:

Related to art - Bug #5831: SimpleMemoryCheck affected by overflow errors Closed

History

#1 - 12/09/2011 04:37 PM - Walter E Brown

- Status changed from New to Accepted

SLF4 is not a concern for us. We ought verify the functionality under both SLF5 and SLF6.

#2 - 08/02/2013 02:00 PM - Christopher Green

- Due date set to 09/30/2013
- Target version set to 1.09.00
- Estimated time set to 20.00 h
- Scope set to Internal
- SSI Package art added

#3 - 02/17/2014 12:27 PM - Christopher Green

- Target version changed from 1.09.00 to 521

#4 - 09/02/2014 02:23 PM - Christopher Green

- Target version changed from 521 to 1.13.00

#5 - 12/19/2014 03:15 PM - Kyle Knoepfel

- Assignee set to Kyle Knoepfel

10/21/2020 1/3

#6 - 12/19/2014 03:48 PM - Kyle Knoepfel

- % Done changed from 0 to 70

The proofs types and fields for SLF5 and SLF6 are shown below. Entries prefaced with '*' indicate either a difference in type or field.

1. pid		Property	SLF5	SLF6
3. state %c %c 4. ppid %d			%d	%d
4. ppid	2.	comm	%s	%s
5. pgrp %d %d %d 6. session %d %d %d 7. tty_nr %d %d %d 8. tpgid %d %d %d * 9. flags %lu %u 10. minflt %lu %lu 11. cminflt %lu %lu %lu 12. majflt %lu %lu %lu 13. cmajflt %lu %lu %lu 14. utime %lu %lu %lu 15. stime %ld %ld 17. cstime %ld %ld 18. priority %ld %ld 19. nice %ld %ld 20. num_threads %ld %ld 21. itrealvalue %ld %ld 22. starttime %lu %lu 23. vsize %lu %lu 24. rss %ld %ld 27. endcode %lu %lu 27. endcode %lu %lu 28. startstack %lu %lu 29. kstkesp %lu %lu 30. kstkeip %lu %lu 31. signal %lu %lu 32. blocked %lu %lu 33. sigignore %lu %lu 34. sigcatch %lu %lu 35. wchan %lu %lu 36. nswap %lu %lu 37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *43. guest_time *10.	3.		%C	%C
5. pgrp %d %d %d 6. session %d %d %d 7. tty_nr %d %d %d 8. tpgid %d %d %d * 9. flags %lu %u 10. minflt %lu %lu %lu 11. cminflt %lu %lu %lu 12. majflt %lu %lu %lu 13. cmajflt %lu %lu %lu 14. utime %lu %lu %lu 15. stime %ld %ld 17. cstime %ld %ld 18. priority %ld %ld 19. nice %ld %ld 20. num_threads %ld %ld 21. itrealvalue %ld %ld 22. starttime %lu %lu 23. vsize %lu %lu 24. rss %ld %ld 25. rlim %lu *25. rlim %lu *25. rsslim %lu 26. startcode %lu %lu 27. endcode %lu %lu 28. startstack %lu %lu 29. kstkesp %lu %lu 30. kstkeip %lu %lu 31. signal %lu %lu 32. blocked %lu %lu 33. sigignore %lu %lu 34. sigcatch %lu %lu 35. wchan %lu %lu 36. nswap %lu %lu 37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *43. guest_time %lu %10. *10. *10. *10. *10. *10. *10. *10. *	4.	ppid	%d	%d
6. session 7. tty_nr 8d %d %d 8. tpgid %d %d %d * 9. flags %lu %u 10. minflt %lu %lu 11. cminflt %lu %lu %lu 12. majflt %lu %lu %lu 13. cmajflt %lu %lu %lu 14. utime %lu %lu %lu 15. stime %ld %ld 17. cstime %ld %ld 18. priority %ld %ld 19. nice %ld %ld 20. num_threads %ld %ld 21. itrealvalue %ld %ld *22. starttime %lu %lu 23. vsize %lu %lu 24. rss %ld %ld	5.		%d	%d
8. tpgid	6.	= = =	%d	%d
* 9. flags	7.	tty_nr	%d	%d
10. minflt	8.	tpqid	%d	%d
10. minflt	* 9.	flags	%lu	%u
12. majflt		minflt	%lu	%lu
13. cmajfit % lu % lu 14. utime % lu % lu 15. stime % lu % lu 16. cutime % ld % ld 17. cstime % ld % ld 18. priority % ld % ld 19. nice % ld % ld 20. num_threads % ld % ld 21. itrealvalue % ld % ld 22. starttime % lu % lu 23. vsize % lu % lu 24. rss % ld % ld 25. rlim % lu *25. rlim % lu *25. rsslim % lu 26. startcode % lu % lu 27. endcode % lu % lu 28. startstack % lu % lu 29. kstkesp % lu % lu 30. kstkeip % lu % lu 31. signal % lu % lu 33. sigignore % lu % lu 33. sigignore % lu % lu 34. sigcatch % lu % lu 35. wchan % lu % lu 36. nswap % lu % lu 37. cnswap % lu % lu 38. exit_signal % d % d 39. processor % d % d 40. rt_priority % lu % u 41. policy % lu % u 42. delayacct_blkio_ticks % llu % llu *43. guest_time % lu	11.	cminflt	%lu	%lu
14. utime 15. stime 16. cutime 16. cutime 17. cstime 18. priority 19. nice 20. num_threads 21. itrealvalue 22. starttime 23. vsize 24. rss 25. rlim 26. startcode 27. endcode 28. startstack 29. kstkesp 30. kstkeip 30. kstkeip 31. signal 32. blocked 33. sigignore 34. sigcatch 35. wchan 36. nswap 38. exit_signal 38. exit_signal 38. exit_signal 39. processor 40. rt_priority 41. policy 41. policy 42. delayacct_blkio_ticks 4ld 4ld 4ld 4ld 4ld 4ld 4lu	12.	majflt	%lu	%lu
14. utime 15. stime 16. cutime 16. cutime 17. cstime 18. priority 19. nice 20. num_threads 21. itrealvalue 22. starttime 23. vsize 24. rss 25. rlim 26. startcode 27. endcode 28. startstack 29. kstkesp 30. kstkeip 30. kstkeip 31. signal 32. blocked 33. sigignore 34. sigcatch 35. wchan 36. nswap 38. exit_signal 38. exit_signal 38. exit_signal 39. processor 40. rt_priority 41. policy 41. policy 42. delayacct_blkio_ticks 4ld 4ld 4ld 4ld 4ld 4ld 4lu	13.	cmajflt	%lu	%lu
16. cutime			%lu	%lu
17. cstime 18. priority 19. nice 20. num_threads 21. itrealvalue *22. starttime 23. vsize 24. rss *1d *1d *1u 25. rlim *25. rlim *26. startcode 27. endcode 28. startstack 29. kstkesp 30. kstkeip 31. signal 31. signal 32. blocked 33. sigignore 34. sigcatch 35. wchan 36. nswap 37. cnswap 38. exit_signal 38. exit_signal 39. processor *40. rt_priority *41. policy 42. delayacct_blkio_ticks *1d *1d *1d *1d *1d *1d *1d *1	15.		%lu	%lu
18. priority	16.	cutime	%ld	%ld
19. nice	17.	cstime	%ld	%ld
20. num_threads 21. itrealvalue	18.	priority	%ld	%ld
21. itrealvalue	19.		%ld	%ld
*22. starttime	20.	num_threads	%ld	%ld
23. vsize	21.	itrealvalue	%ld	%ld
23. vsize 24. rss *lu %lu 24. rss *ld %ld *25. rlim *25. rsslim 26. startcode *lu %lu 27. endcode 28. startstack 29. kstkesp 30. kstkeip 31. signal 32. blocked 33. sigignore 34. sigcatch 35. wchan 36. nswap 37. cnswap 38. exit_signal 38. exit_signal 39. processor *d *d *d *d *d *d *d *d *d *	*22.	starttime	%lu	%llu
24. rss %ld %ld				
*25. rlim	23.	vsize	%lu	%lu
*25. rsslim	24.	rss	%ld	%ld
*25. rsslim				
26. startcode				
27. endcode %lu %lu 28. startstack %lu %lu 29. kstkesp %lu %lu 30. kstkeip %lu %lu 31. signal %lu %lu 32. blocked %lu %lu 33. sigignore %lu %lu 34. sigcatch %lu %lu 35. wchan %lu %lu 36. nswap %lu %lu 37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *41. policy %lu %u *42. delayacct_blkio_ticks %llu %llu *43. guest_time %lu				
28. startstack	26.	startcode	%lu	%lu
29. kstkesp %lu %lu 30. kstkeip %lu %lu 31. signal %lu %lu 32. blocked %lu %lu 33. sigignore %lu %lu 34. sigcatch %lu %lu 35. wchan %lu %lu 36. nswap %lu %lu 37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time %lu				
30. kstkeip %lu %lu 31. signal %lu %lu 32. blocked %lu %lu 33. sigignore %lu %lu 34. sigcatch %lu %lu 35. wchan %lu %lu 36. nswap %lu %lu 37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time ~ %lu		startstack		
31. signal %lu %lu 32. blocked %lu %lu 33. sigignore %lu %lu 34. sigcatch %lu %lu 35. wchan %lu %lu 36. nswap %lu %lu 37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time ~ %lu		kstkesp		
32. blocked %lu %lu 33. sigignore %lu %lu 34. sigcatch %lu %lu 35. wchan %lu %lu 36. nswap %lu %lu 37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time %lu		kstkeip		
33. sigignore %lu %lu 34. sigcatch %lu %lu 35. wchan %lu %lu 36. nswap %lu %lu 37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time %lu		-		
34. sigcatch %lu %lu 35. wchan %lu %lu 36. nswap %lu %lu 37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time %lu	32.	blocked	%lu	%lu
35. wchan %lu %lu 36. nswap %lu %lu 37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time ~ %lu				
36. nswap %lu %lu 37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time ~ %lu		sigcatch		%lu
37. cnswap %lu %lu 38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time %lu	35.	wchan	%lu	%lu
38. exit_signal %d %d 39. processor %d %d *40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time %lu		nswap		
39. processor %d %d %d *40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time %lu				%lu
*40. rt_priority %lu %u *41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time %lu		exit_signal	%d	%d
*41. policy %lu %u 42. delayacct_blkio_ticks %llu %llu *43. guest_time %lu		-		%d
42. delayacct_blkio_ticks %llu %llu *43. guest_time %lu	*40.	rt_priority	%lu	%u
*43. guest_time %lu	*41.	-		
		delayacct_blkio_ticks	%llu	%llu
*44. cguest_time %ld		=		
	*44.	cguest_time		%ld

For the current memory service, only items 23 and 24 are of interest. The code is being restructured so that only the values of fields 23 and 24 are stored. This is done by a new function that looks like:

```
vsize_t vsize;
rss_t rss;
std::istringstream iss ( buffer_from_proc_stat );
iss >> token_ignore(22) >> vsize >> rss;
```

Should a user decide they want the values associated with any other fields, we will need to come up with an SL-release-dependent way of handling this. For now, this is not a concern.

The types of vsize and rss in the current version of SimpleMemoryCheck_service.cc are being corrected to be in accord with those provided by SL5/6 procfs.

#7 - 12/22/2014 09:10 AM - Kyle Knoepfel

- Status changed from Accepted to Assigned

10/21/2020 2/3

#8 - 01/09/2015 10:22 AM - Kyle Knoepfel

- Status changed from Assigned to Resolved
- % Done changed from 70 to 100

This has been implemented with fbc282dd.

#9 - 02/16/2015 10:32 AM - Christopher Green

- Status changed from Resolved to Closed

10/21/2020 3/3